

What is claimed is:

1. A method for programmatically enforcing referential integrity constraints among associations between class instances, comprising steps of:  
evaluating a request to set an association end to reflect an association from an instance of a first class to an instance of a second class;  
setting the requested association end; and  
programmatically modifying an inverse association end of the association to reflect an inverse association from the instance of the second class to the instance of the first class.

2. The method according to Claim 1, wherein the evaluating step further comprises determining whether the association end has a single multiplicity or a many multiplicity.

3. The method according to Claim 2, wherein the setting and programmatically modifying steps for a particular association end that has a single multiplicity further comprise steps of:  
disconnecting the inverse association end from an existing instance, if any;  
performing the programmatically modifying step after performing the disconnecting step; and  
performing the setting step after performing the disconnecting step.

4. The method according to Claim 2, wherein the setting and programmatically modifying steps for a particular association end that has a many multiplicity further comprise steps of:  
performing the setting step;

4           disconnecting the inverse association end from an existing instance, if any, after  
5 performing the setting step; and  
6           performing the programmatically modifying step after performing the setting step.

1       5.     The method according to Claim 1, further comprising steps of:  
2           determining whether the association end or the inverse association end is a primary end  
3 of the association; and  
4           serializing only the primary end of the association during a serialization operation.

6       6.     The method according to Claim 1, wherein the method is provided as link helper  
2 objects.

1       7.     A computer program product for programmatically enforcing referential integrity  
2 constraints among associations between class instances, wherein the computer program  
3 product is embodied on one or more computer readable media and comprises:

4           computer-readable program code means for evaluating a request to set an association  
5 end to reflect an association from an instance of a first class to an instance of a second class;  
6           computer-readable program code means for setting the requested association end; and  
7           computer-readable program code means for programmatically modifying an inverse  
8 association end of the association to reflect an inverse association from the instance of the  
9 second class to the instance of the first class.

1 8. The computer program product according to Claim 7, wherein the computer-readable  
2 program code means for evaluating further comprises computer-readable program code means  
3 for determining whether the association end has a single multiplicity or a many multiplicity.

1 9. The computer program product according to Claim 8, wherein the computer-readable  
2 program code means for setting and the computer-readable program code means for  
3 programmatically modifying for a particular association end that has a single multiplicity  
4 further comprise:

5 computer-readable program code means for disconnecting the inverse association end  
6 from an existing instance, if any;

7 computer-readable program code means for operating the computer-readable program  
8 code means for programmatically modifying after operating the computer-readable program  
9 code means for disconnecting; and

10 computer-readable program code means for operating the computer-readable program  
11 code means for setting after operating the computer-readable program code means for  
12 disconnecting; and

13 wherein the computer-readable program code means for setting and the computer-readable  
14 program code means for programmatically modifying for a particular association end that has  
15 a many multiplicity further comprise:

16 computer-readable program code means for performing the setting;

17 computer-readable program code means for disconnecting the inverse association end  
18 from an existing instance, if any, after operation of the computer-readable program code

19 means for setting; and  
20 computer-readable program code means for operating the computer-readable program  
21 code means for programmatically modifying after operating the computer-readable program  
22 code means for setting.

1 11. The computer program product according to Claim 7, further comprising:  
2 computer-readable program code means for determining whether the association end  
3 or the inverse association end is a primary end of the association; and  
4 computer-readable program code means for serializing only the primary end of the  
5 association during a serialization operation.

6 12. A system for programmatically enforcing referential integrity constraints among  
7 associations between class instances, comprising:  
8 means for evaluating a request to set an association end to reflect an association from  
9 an instance of a first class to an instance of a second class;  
10 means for setting the requested association end; and  
11 means for programmatically modifying an inverse association end of the association to  
12 reflect an inverse association from the instance of the second class to the instance of the first  
13 class.

1 13. The system according to Claim 12, wherein the means for evaluating further comprises  
2 means for determining whether the association end has a single multiplicity or a many

3 multiplicity.

1 14. The system according to Claim 13, wherein the means for setting and the means for  
2 programmatically modifying for a particular association end that has a single multiplicity  
3 further comprise:

4 means for disconnecting the inverse association end from an existing instance, if any;

5 means for operating the means for programmatically modifying after operating the  
6 means for disconnecting; and

7 means for operating the means for setting after operating the means for disconnecting;  
8 and

9 wherein the means for setting and the means for programmatically modifying for a particular  
10 association end that has a many multiplicity further comprise:

11 means for performing the setting;

12 means for disconnecting the inverse association end from an existing instance, if any,  
13 after operation of the means for setting; and

14 means for operating the means for programmatically modifying after operating the  
15 means for setting.

1 15. The system according to Claim 11, further comprising:

2 means for determining whether the association end or the inverse association end is a  
3 primary end of the association; and

4 means for serializing only the primary end of the association during a serialization

